CLAIMS:

What is claimed is:

1. A method of debugging a software program, said software program executing properly with at least one of a first set of options and executing improperly with at least one of a second set of options, said method comprising the steps of:

generating a first log file by executing said program with said at least one of said first set of options, said first log file including all functions executed by said program during this first execution;

generating a second log file by executing said program with said at least one of said second set of options, said second log file including all functions executed by said program during this second execution; and

comparing said first log file with said second log file to debug the software program.

2. The method according to claim 1, further comprising the steps of:

generating said first log file including a first set of return codes;

generating said second log file including a second set of return codes; and

comparing said first set of return codes with said second set of return codes to debug the software program.

3. The method according to claim 1, further comprising the step of compiling said software program to generate

compiled code, said compiled code including a listing of said functions.

4. The method according to claim 3, further comprising the step of compiling said software program utilizing a C compiler and utilizing a "-g" option, said "-g" option generating said listing of said function.

5. The method according to claim 3, further comprising the step of generating a file including said listing obtained from said compiled code.

6. The method according to claim 5, further comprising the step of utilizing a UNIX dump command to generate said file, said UNIX dump command causing an output of said listing.

7. The method according to claim 6, further comprising the step of generating a debug script utilizing said file.

8. The method according to claim 1, further comprising the steps of:

automatically generating a debug script including the steps of:

generating script code for each of a plurality of function calls included in said software program, said script code setting a breakpoint at each of said plurality of function calls;

generating script code which logs each of a plurality of said plurality of functions calls executed by said software program when

said software program is executed under the control of said debug program; and

generating script code which causes execution of said software program to continue after each of said plurality of said plurality of function calls is logged.

9. A computer program product for debugging a software program, said software program executing properly with at least one of a first set of options and executing improperly with at least one of a second set of options, said computer program product comprising:

instruction means for generating a first log file by executing said program with said at least one of said first set of options, said first log file including all functions executed by said program during this first execution;

instruction means for generating a second log file by executing said program with said at least one of said second set of options, said second log file including all functions executed by said program during this second execution; and

instruction means for comparing said first log file with said second log file to debug the software program.

10. The product according to claim 9, further comprising:

instruction means for generating said first log file including a first set of return codes;

instruction means for generating said second log file including a second set of return codes; and

instruction means for comparing said first set of return codes with said second set of return codes to debug the software program.

11. The product according to claim 9, further comprising instruction means for compiling said software program to generate compiled code, said compiled code including a listing of said functions.

12. The product according to claim 11, further comprising instruction means for compiling said software program utilizing a C compiler and utilizing a "-g" option, said "-g" option generating said listing of said function.

13. The product according to claim 11, further comprising instruction means for generating a file including said listing obtained from said compiled code.

14. The product according to claim 13, further comprising instruction means for utilizing a UNIX dump command to generate said file, said UNIX dump command causing an output of said listing.

15. The product according to claim 14, further comprising instruction means for generating a debug script utilizing said file.

16. The product according to claim 9, further comprising:

instruction means for automatically generating a debug script including:

instruction means for generating script code for each of a plurality of function calls included in said software program, said script code setting a breakpoint at each of said plurality of function calls;

instruction means for generating script code which logs each of a plurality of said plurality of functions calls executed by said software program when said software program is executed under the control of said debug program; and

instruction means for generating script code which causes execution of said software program to continue after each of said plurality of said plurality of function calls is logged.

17. A system for debugging a software program, said software program executing properly with at least one of a first set of options and executing improperly with at least one of a second set of options, comprising:

a first log file being generated by executing said program with said at least one of said first set of options, said first log file including all functions executed by said program during this first execution;

a second log file being generated by executing said program with said at least one of said second set of options, said second log file including all functions executed by said program during this second execution; and

means for comparing said first log file with said second log file to debug the software program.

18. The system according to claim 17, further comprising:

said first log file being generated including a first set of return codes;

said second log file being generated including a second set of return codes; and

means for comparing said first set of return codes with said second set of return codes to debug the software program.

19. The system according to claim 17, further comprising said software program being compiled to generate compiled code, said compiled code including a listing of said functions.

20. The system according to claim 19, further comprising said software program being compiled utilizing a C compiler and utilizing a "-g" option, said "-g" option generating said listing of said function.

21. The system according to claim 19, further comprising a file being generated including said listing obtained from said compiled code.

22. The system according to claim 21, further comprising a UNIX dump command being utilized to generate said file, said UNIX dump command causing an output of said listing.

23. The system according to claim 22, further comprising a debug script being generated utilizing said file.

24. The system according to claim 17, further comprising:

a debug script being automatically generated including:

script code being generated for each of a plurality of function calls included in said software program, said script code setting a breakpoint at each of said plurality of function calls;

script code being generated which logs each of a plurality of said plurality of functions calls executed by said software program when said software program is executed under the control of said debug program; and

script code being generated which causes execution of said software program to continue after each of said plurality of said plurality of function calls is logged.